PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L | A2

(11) International Publication Number: **WO 99/52238**

(43) International Publication Date: 14 October 1999 (14.10.99)

(21) International Application Number: PCT/US99/04928

(22) International Filing Date: 5 March 1999 (05.03.99)

(30) Priority Data:
09/053,851 2 April 1998 (02.04.98) US

(71) Applicant: MOTOROLA INC. [US/US]; 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

(72) Inventor: LOCKHART, Thomas, Wayne; 4580 Maple Crescent, Vancouver, British Columbia V6J 4B4 (CA).

(74) Agents: NICHOLS, Daniel, K. et al.; Motorola Inc., Intellectual Property Dept., 1500 Gateway Boulevard, Boynton Beach, FL 33426-8292 (US).

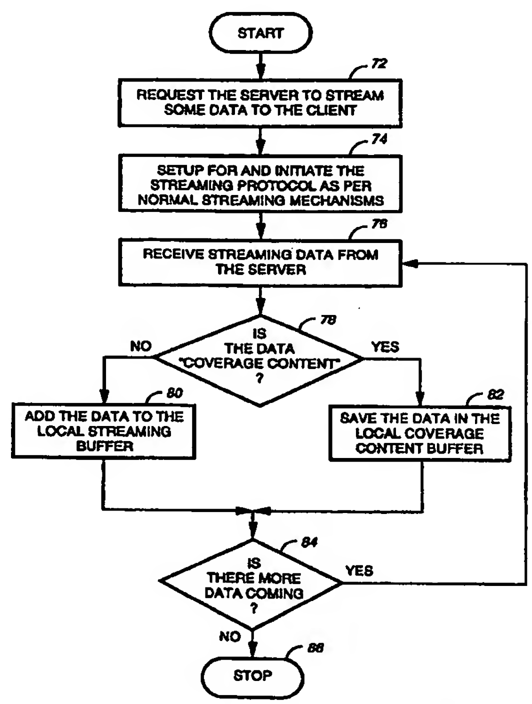
(81) Designated States: AU, CA, CN, JP, MX, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: METHOD AND APPARATUS FOR GAP COVERAGE IN STREAMING PROTOCOLS

(57) Abstract

A data streaming client (20) providing gap coverage with data st reaming protocols includes a processor (22) and a memory (24) for storing requested data requested by the data streaming client and a memory (28) for storing gap data. The processor is preferably programmed to receive streaming data (76) including requested data (12) and appropriate gap data (14) from a server (16). The processor should also be able to distinguish the requested data from the appropriate gap data (78) and to store the requested data to a first buffer (24) and store the appropriate gap data to a second buffer (28).



20

METHOD AND APPARATUS FOR GAP COVERAGE IN STREAMING PROTOCOLS

FIELD OF THE INVENTION

The present invention is directed to a communication protocol, and more particularly to a communication device and method capable of providing data content to a client during gaps in presenting data streaming protocols.

BACKGROUND OF THE INVENTION

When using an internet data streaming protocol or any data streaming protocol, there are often time gaps or delays in the delivery of data to the viewer. The data could consist of text, but could also include video or audio material. These gaps are currently unused and waste time for the user or viewer to the point that the user or viewer may lose interest.

The data streaming protocol initially used on the internet's World Wide Web (WWW) for communicating audio, video, or similar information provided for a server to pass a file containing the data to the client. After receiving the entire file, the client would play or display the file. This mechanism was very slow, often involving many minutes of delay until the file was received by the client. Once the files were received, however, they could be played in real time with few pauses or gaps. "Data streaming" was created to address this excessive delay experienced, particularly when receiving multimedia content. Basically, data streaming works by dividing the real time data into packets, which are sent from the server to the client. The client buffers a number of the packets, (the number based on expected delays in the delivery of subsequent packets) and then starts to play the data before it has received the entire data set (video, audio or like file). While the data is playing or being presented, the remainder of the data is received. This mechanism drastically reduces the initial delay experienced by the user of the system.

This streaming mechanism does however create a new problem. Namely, that the subsequent packets may not arrive in a timely fashion as expected, resulting in gaps in the resulting presentation to the user. In other words, the client did not buffer enough data prior to beginning the presentation causing the client to run out of data to present to the user because the client failed to receive the rest of the data soon enough. Existing streaming mechanisms simply stop and leave the user waiting until

there is enough data to proceed with the presentation. Existing data streaming mechanisms such as Real-Time Streaming Protocol (RTSP) used by RealAudio and the Real-Time Transport Protocol (RTP) IETF standard which it sits atop work roughly in the same fashion as just described. Thus, a need exists to make effective use of these gaps in the delivery of streaming data.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a system providing gap coverage for internet streaming protocols in accordance with the present invention.

FIG. 2 is a flow chart illustrating a method for providing gap coverage at a data streaming server in accordance with the present invention.

FIG. 3 is a flow chart illustrating a method for receiving gap coverage data at a data streaming client receiver in accordance with the present invention.

FIG. 4 is a flow chart illustrating a method for presenting gap coverage data at a data streaming client receiver unit in accordance with the present invention.

DETAILED DESCRIPTION

The data streaming protocol used to communicate the real-time data 12 (such as a video source) from the server to the client is enhanced to identify and carry a new type of content. This new content is the "coverage content" 14 or gap data which is likely an advertisement of some sort. The details of the enhancements required would depend upon the streaming protocol to be used and the exact nature of the coverage content.

Referring to FIG. 1, a system that provides gap coverage in data streaming protocols in accordance with the present invention is shown. Preferably, the system 10 comprises a data streaming server 16 that has a memory 15 for storing requested data requested by a client and for storing gap data and a processor 17. Preferably the processor 17 is programmed to receive a request from a client 20 to stream requested data to the client. The processor 17 would also be programmed to set up and initiate a streaming protocol. The processor 17 may also select appropriate gap data that can be streamed at a predetermined time during the data streaming protocol. The

predetermined time is preferably at the beginning of the data stream. Additionally, the processor 17 formats the appropriate gap data. Once this is done, the appropriate gap data can be sent during the predetermined time. Of course, the processor 17 would preferably be programmed to send and complete the requested data to the client after the gap data has been sent. The transmission of data between the server 16 and client 20 is preferably done over a data network 18 such as the internet.

The server would insert the identified coverage content (i.e., ads), possibly several instances thereof, into the data stream at various points. The very beginning of the stream would be a good place, as well as after significant amounts of data had been streamed. The nature of the coverage content could be simple pure text messages, still graphic images, audio files, short video clips or the like. The type of coverage would be selected to be compatible with the data being streamed. For example, video clips would not be suitable coverage content for streamed audio, but they might be for streamed full length movies.

A data streaming client 20 preferably comprises a memory 24 for storing requested data requested by the data streaming client and a memory 28 for storing gap data. The data streaming client 20 would also comprise a processor (22 and 26) that was programmed to request from a server requested data to the data streaming client, receive the requested data and appropriate gap data from the server, distinguish the requested data from the appropriate gap data, and store the requested data to a first buffer (24) and store the appropriate gap data to a second buffer (28). Ideally, the processor is further programmed to present the contents of the second buffer while the contents of the first buffer is waiting to be filled to the point when the contents of the first buffer can be presented.

The client would receive the data stream and strip off the coverage content, saving it in a local coverage content buffer. It would then proceed to receive, buffer, and deliver the streamed data as per the normal streaming mechanisms. If the streaming buffer (located in the client) ever became empty (or close to empty), the client would replace the normal streamed output with an appropriate representation of the coverage content (extracted from the client's local coverage content buffer), until such time as the streaming buffers refilled to the point where reliable normal streaming delivery could be continued. It should be understood that the client's local

coverage content buffer (the second buffer 28) can be filled with appropriate gap data that may come from the server or alternatively locally provided at the client when insufficient or no gap data is streamed from the server to the client.

For example, if the client had requested and was streaming a movie from the server, then the coverage content might consist of one or several graphics which were ads for some product (other movies perhaps). In other words, the processor 26 can be programmed to convert the contents of the second buffer to a format compatible with the format of the contents of the first buffer. If the streaming buffers became very low, due to delays in the delivery of the streamed data, the client would stop presenting the movie and present the coverage content graphics (in appropriate video format) to the user. The streaming data would (hopefully) continue to arrive and begin to refill the local streaming buffers. When these buffers had refilled to the point where it was appropriate to continue the movie, the client would replace the coverage content with the movie again and go back to normal data streaming processing.

Of course, several variations are contemplated with the scope of the claims of the present invention. For instance, a client capable of the gap coverage mechanism in the present invention could include canned coverage content such as an ad for the client if no suitable coverage content was received from the server. Typically, this canned coverage content would be generated locally at the client. The data stream may include different coverage content items which could be used by the client based upon various algorithms including last received, cycle through, or some other mechanism coordinated with the server. Additionally, the data streaming protocol could include "markers" of good locations where to include ads if their buffers were low, which would allow the ad insertion to happen at possibly more acceptable points in the program (an intermission in a movie for instance). Advertisements could be presented during the initial delay while waiting for a client's local streaming buffer to fill to the point where presentation of the real signal can start. Another feature within the scope of the present invention would allow coverage content material represented as text to be converted to either or both audio and video signals by the client as needed. Further, coverage content or gap data could be filtered by the client based on location, user preferences, remaining battery life, or other considerations. In other

words, the processor at the client can be programmed to filter the contents of the second buffer based on criteria such as location of the client, user preferences, and remaining battery life (particularly if the client was a portable device).

FIG. 2 illustrates a method 50 at a server for providing gap coverage in data streaming protocols. The method 50 preferably begins at step 52 where the server receives a request from a client to stream requested data to the client. The server at step 54 then sets up and initiates a data stream. Setting up and initiating the streaming the data could include providing markers where the appropriate gap data can be placed within the data stream. At step 56, the server selects appropriate gap data that can be streamed at predetermined times during the data stream. The appropriate gap data could include canned coverage data that is programmed to be sent with particular kinds of data requests. Then, the server formats and sends the appropriate gap data at step 58 during the predetermined times which preferably is at the beginning of the data stream or after large amounts of data had already been sent. Finally, the server continues to send and complete the requested data to the client at step 60.

FIGs. 3 and 4 illustrate a method 70 for receiving and a method for presenting 90 appropriate gap data in data streaming protocols at a client. Preferably, the method 70 comprises the step of requesting from a server to stream requested data to the client at step 72 and then setting up and initiating the data stream as would typically be done for normal streaming mechanisms at step 74. The client at step 76 would then receive the requested data and appropriate gap data from the server. At decision block 78, the client would need to distinguish between the requested data and the appropriate gap data. If the data received at decision block 78 was the requested data, then the requested data would be stored at a first buffer at step 80. If the data received at decision block 78 was the appropriate gap data, then the appropriate gap data would be stored at a second buffer at step 82. Of course, the method 70 would continue to look for more data at step 84 until no more data was received.

Referring to FIG. 4, a method 90 for presenting the appropriate gap data is shown. At step 92, the client waits for a request for streaming data to be sent to a server. A server would receive the request and stream data back to the client as

previously explained above. At step 94, the client would wait for the first buffer or local streaming buffer to fill up to the point where there is enough data to begin a presentation to the user of the client. Next, the method would continue by removing a block of data from the local streaming buffer and format it for presentation to the user at step 96. At decision block 98, a determination is made whether the presentation is completed. If completed, then the process stops at step 106. If the presentation is not completed, then another inquiry is made to determine whether the local streaming buffer (the first buffer) contents was below a cutoff level at decision block 100. If the contents is not below the cutoff level, then another block of data is removed from the local streaming buffer and subsequently formatted as shown in step 96. If the contents is below the cutoff level, then the method proceeds to step 102 by obtaining the coverage content (appropriate gap data) from the local coverage content buffer (the second buffer) and formatting and presenting such coverage content. It should be understood that formatting the coverage content may include converting the contents of the second buffer to a format compatible with the format of the contents of the first buffer. Once again, an inquiry is made at decision block 104 to determine whether contents of the first buffer was above or below the cutoff level. If sufficient data is found in the local streaming buffer, then the method returns to step 96. If insufficient data is found in the local streaming buffer, then the method obtains further coverage content at step 102.

The above description is intended by way of example only and is not intended to limit the present invention in any way except as set forth in the following claims.

What is claimed is:

1. A method for providing gap coverage in a data streaming protocol, comprising the steps at a server of:

receiving a request from a client to stream requested data to the client;

setting up and initiating a data stream;

selecting appropriate gap data that can be streamed at a predetermined time during the data stream;

formatting the appropriate gap data; and

sending the appropriate gap data during the predetermined time.

2. The method of claim 1, wherein the step of selecting appropriate gap data comprises the step of selecting canned coverage data.

3. The method of claim 1, wherein the method further comprises the step of providing markers within the data stream indicating when the appropriate gap data should be preferably decoded by the client.

4. The method of claim 1, wherein the step of sending the appropriate gap data during the predetermined time further comprises the step of sending the appropriate gap data at the beginning of the data stream.

5. A method for receiving appropriate gap data in data streaming protocols, comprising the steps at a client of:

requesting from a server to stream requested data to the client;

receiving the requested data and appropriate gap data from the server;

distinguishing the requested data from the appropriate gap data; and

storing the requested data to a first buffer and storing the appropriate gap data to a second buffer.

6. The method of claim 5, wherein the method further comprises the step of presenting the contents of the second buffer while the contents of the first buffer is waiting to be filled to the point when the contents of the first buffer can be presented.

7. The method of claim 6, wherein the method further comprises the step of converting the contents of the second buffer to a format compatible with the format of the contents of the first buffer.

8. The method of claim 6, wherein the method further comprises the step of filtering the contents of the second buffer before the step of presenting based on criteria selected from the group consisting of location of the client, user preferences, and remaining battery life.

9. A data streaming server that provides gap coverage in a data streaming protocol and communicates with clients through a data network, comprising:

memory for storing requested data requested by a client and for storing gap data; and

a processor programmed to:

receive a request from the client to stream requested data to the client;

set up and initiate a data stream;

select appropriate gap data that can be streamed at a predetermined time during streaming of the data stream;

format the appropriate gap data; and

send the appropriate gap data during the predetermined time.

10. A data streaming client that provides gap coverage in data streaming protocols, comprising:

memory for storing requested data requested by the data streaming client and for storing gap data; and

a processor programmed to:

request to stream from a server requested data to the data streaming client;

receive the requested data and appropriate gap data from the server;

distinguish the requested data from the appropriate gap data; and

store the requested data to a first buffer and store the appropriate gap data to a second buffer.

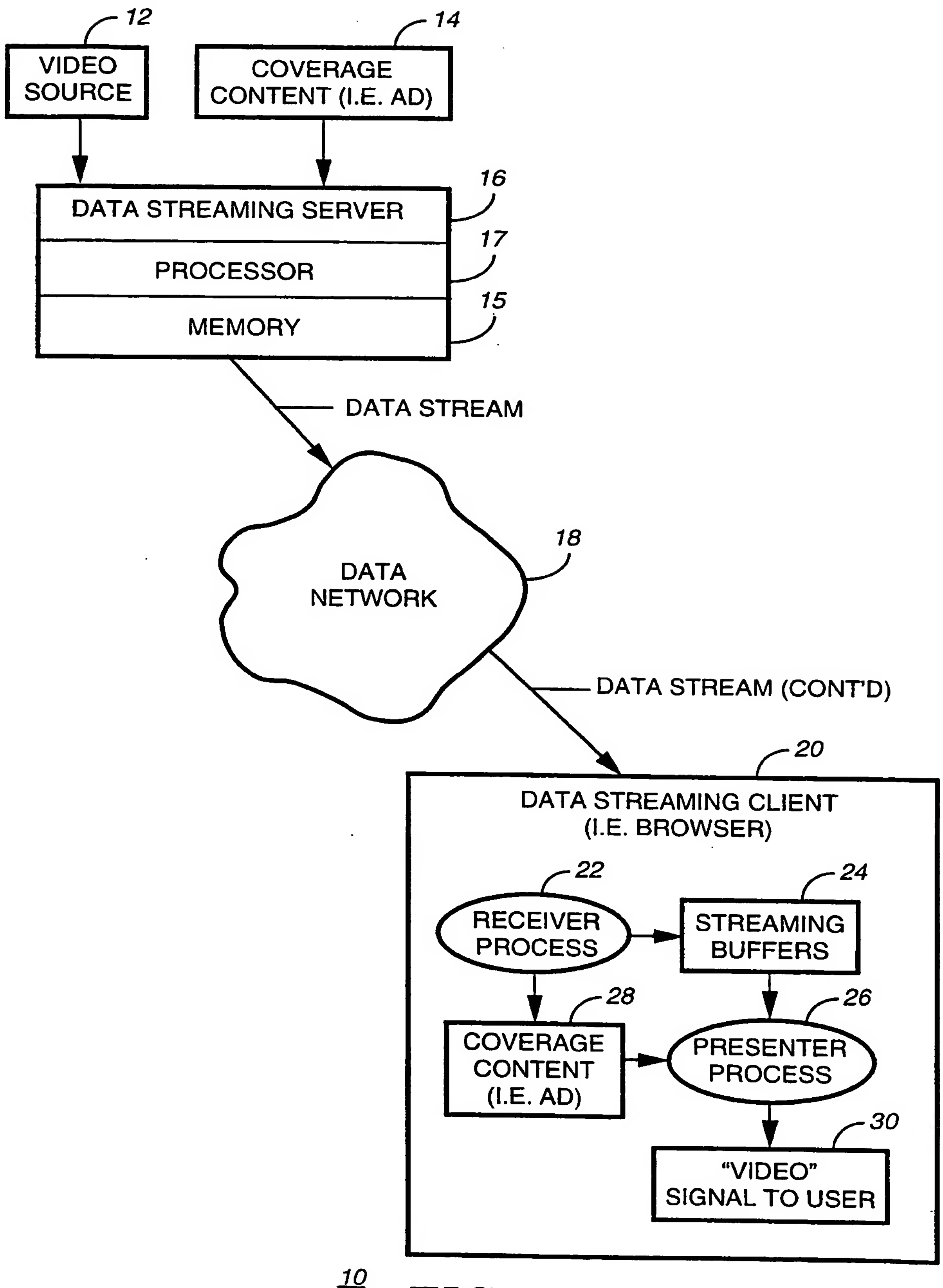

12
VIDEO SOURCE
14
COVERAGE CONTENT (I.E. AD)
16
DATA STREAMING SERVER
17
PROCESSOR
15
MEMORY
DATA STREAM
18
DATA NETWORK
DATA STREAM (CONT'D)
20
DATA STREAMING CLIENT (I.E. BROWSER)
22
RECEIVER PROCESS
24
STREAMING BUFFERS
28
COVERAGE CONTENT (I.E. AD)
26
PRESENTER PROCESS
30
"VIDEO" SIGNAL TO USER
10

FIG. 1

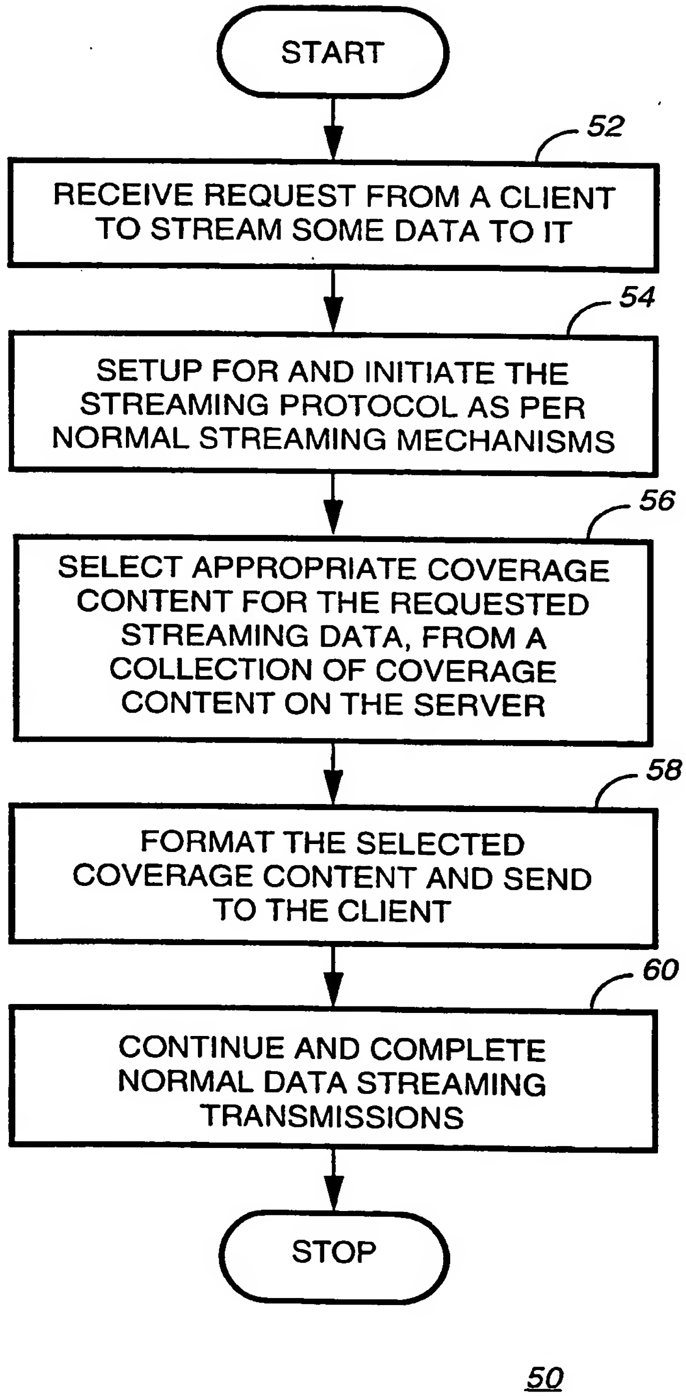

START
52
RECEIVE REQUEST FROM A CLIENT TO STREAM SOME DATA TO IT
54
SETUP FOR AND INITIATE THE STREAMING PROTOCOL AS PER NORMAL STREAMING MECHANISMS
56
SELECT APPROPRIATE COVERAGE CONTENT FOR THE REQUESTED STREAMING DATA, FROM A COLLECTION OF COVERAGE CONTENT ON THE SERVER
58
FORMAT THE SELECTED COVERAGE CONTENT AND SEND TO THE CLIENT
60
CONTINUE AND COMPLETE NORMAL DATA STREAMING TRANSMISSIONS
STOP
50

FIG. 2

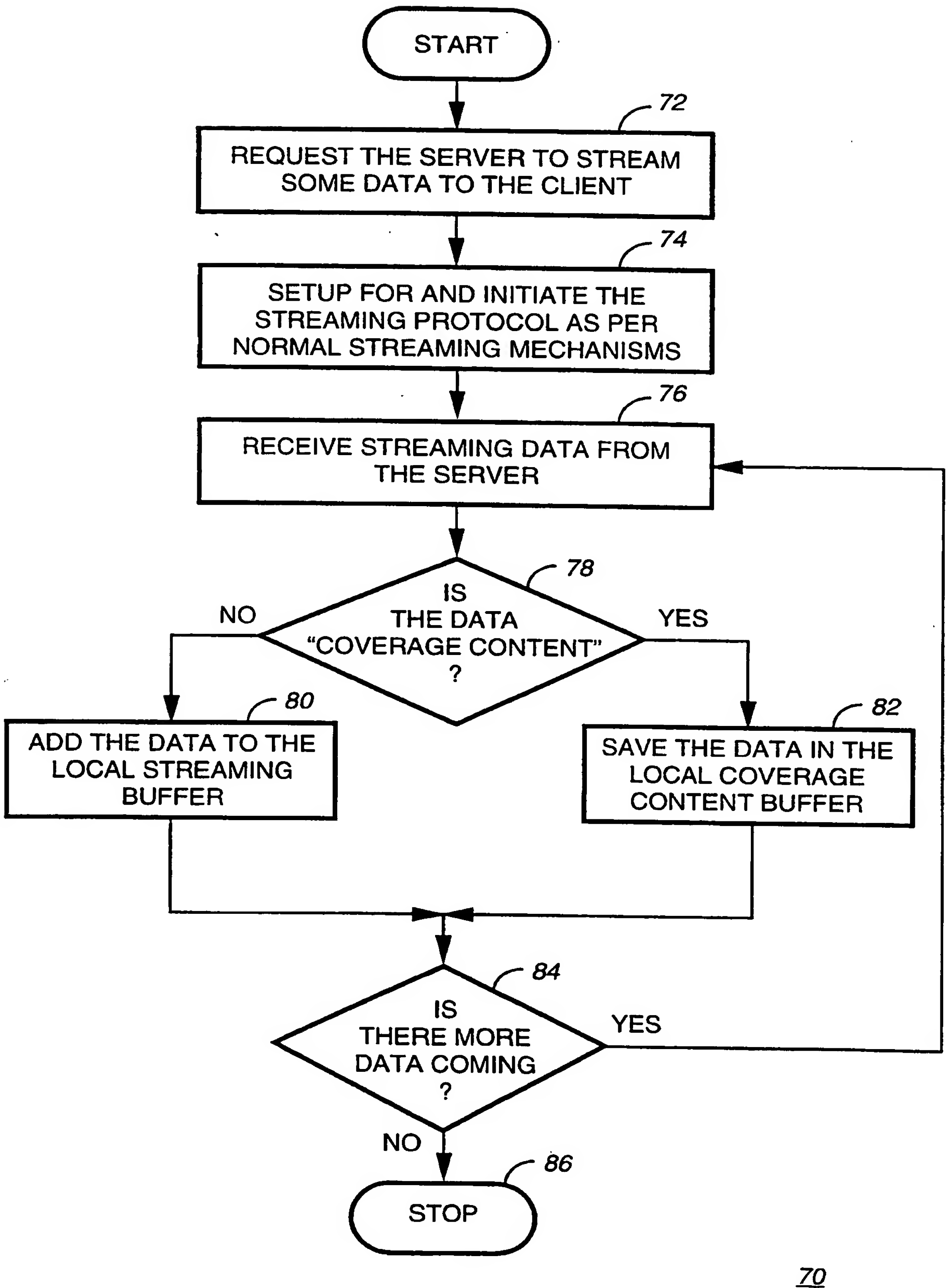
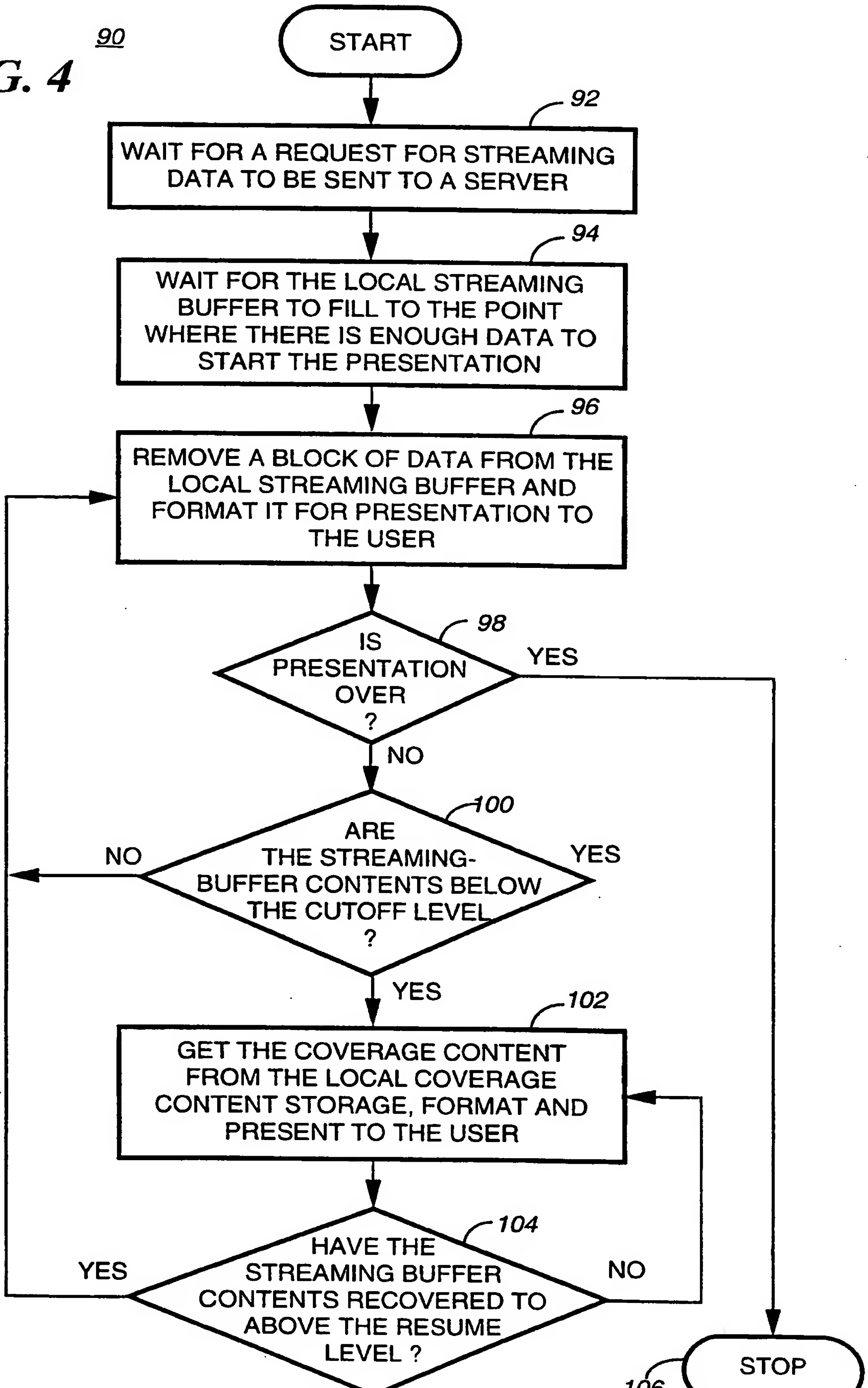

START
72
REQUEST THE SERVER TO STREAM SOME DATA TO THE CLIENT
74
SETUP FOR AND INITIATE THE STREAMING PROTOCOL AS PER NORMAL STREAMING MECHANISMS
76
RECEIVE STREAMING DATA FROM THE SERVER
78
IS THE DATA "COVERAGE CONTENT" ?
NO
YES
80
ADD THE DATA TO THE LOCAL STREAMING BUFFER
82
SAVE THE DATA IN THE LOCAL COVERAGE CONTENT BUFFER
84
IS THERE MORE DATA COMING ?
YES
NO
86
STOP
70

FIG. 3

FIG. 4


90
START
92
WAIT FOR A REQUEST FOR STREAMING DATA TO BE SENT TO A SERVER
94
WAIT FOR THE LOCAL STREAMING BUFFER TO FILL TO THE POINT WHERE THERE IS ENOUGH DATA TO START THE PRESENTATION
96
REMOVE A BLOCK OF DATA FROM THE LOCAL STREAMING BUFFER AND FORMAT IT FOR PRESENTATION TO THE USER
98
IS PRESENTATION OVER ?
YES
NO
100
ARE THE STREAMING-BUFFER CONTENTS BELOW THE CUTOFF LEVEL ?
NO
YES
YES
102
GET THE COVERAGE CONTENT FROM THE LOCAL COVERAGE CONTENT STORAGE, FORMAT AND PRESENT TO THE USER
104
HAVE THE STREAMING BUFFER CONTENTS RECOVERED TO ABOVE THE RESUME LEVEL ?
YES
NO
STOP
106